A Modular Ontology Stack for Integrating OpenBIM and Bayesian Structural Health Monitoring and Prediction

Cedric Driesen[1,*]

[1] Buildwise, Kleine Kloosterstraat 23, 1932 Zaventem, Belgium

Abstract

The Architecture, Engineering, and Construction (AEC) industry faces persistent challenges in integrating heterogeneous data for structural health monitoring (SHM) and facility management, specifically in capturing time-dependent degradation, probabilistic models reflecting uncertainty, and damage semantics. This paper proposes a Linked Data framework that extends Industry Foundation Classes (IFC) with semantic web technologies to address these gaps. By converting IFC data into a Resource Description Framework (RDF) graph via modular ontologies—including BOT, SOSA/SSN, OPM, DOT, and a custom LifeMACS Ontology (LFM)—the framework enables rich semantic queries, provides inputs for probabilistic analyses, and cross-domain interoperability. A Python-based conversion pipeline automates the translation of IFC geometry, sensor data, and inspection records into an RDF knowledge graph, while SPARQL queries demonstrate advanced capabilities, such as probabilistic crack-length or sensor value assessments and real-time data integration. A case study on a deteriorating reinforced concrete bridge validates the approach, showcasing its potential for improved statistical parsing directly using the RDF graph. Results highlight the framework's potential to connect static infrastructure data with dynamic lifecycle parameters. The work highlights the potential of semantic web technologies in advancing digital twin capabilities for the AEC industry.

Keywords

Linked Data, Structural Health, Bayesian Modelling, Probabilistic Modelling, OpenBIM.

1. Introduction

The Architecture, Engineering, and Construction (AEC) industry increasingly relies on digital technologies like Building Information Modelling (BIM) to address the complexities of designing, building, and maintaining infrastructure [1, 2]. OpenBIM—particularly through the Industry Foundation Classes (IFC)—has emerged as a prominent standard for exchanging building and infrastructure information, helping to alleviate the persistent issue of data silos [3]. While direct export from BIM authoring tools to Resource Description Framework (RDF) is possible, leveraging the IFC standard allows the framework to integrate with diverse existing workflows and tools without requiring specific adapters, despite potential information losses inherent in the format conversion from native to IFC. However, despite IFC's widespread adoption, significant challenges remain in capturing rich semantics, facilitating advanced queries, and enabling flexible data linking across heterogeneous sources, especially over a structure's entire life cycle [4].

A range of research efforts has explored ways to augment IFC with semantic web technologies. While direct IFC-to-LBD conversions using schemas like ifcOWL exist [5], integrating domain-specific semantics for SHM often requires mapping to broader Linked Building Data (LBD) ontologies [6–9]. Nonetheless, many existing solutions still struggle with more specialized requirements, such as

LDAC 2025: 13th Linked Data in Architecture and Construction Workshop, July 09–11, 2025, Porto, Portugal

* Corresponding author.

cedric.driesen@buildwise.be

0000-0002-0294-4756

damage representation [10], time-dependent degradation [8], and probabilistic models [11], which are increasingly relevant for structural health monitoring (SHM) applications [12].

This situation becomes especially pressing in the context of facility management, where efficient data integration and reliable interoperability play critical roles. Monitoring concrete structures, for instance, requires aggregating inspection results, sensor measurements, and predictive simulation data—often from multiple specialized tools—while also managing time- and context-sensitive information, such as evolving damage or degradation along with associated uncertainties [12]. Conventional IFC-centric workflows provide robust geometry and basic semantic capabilities but can fall short in representing richer relationships, intricate damage modelling [13, 14], and real-time sensor data stream support [15].

Against this backdrop, Semantic Web Technologies (SWT) and Linked Data (LD) approaches offer an appealing solution to extend traditional BIM data with graph-based models and ontologies. By converting IFC data to RDF and integrating domain-specific ontologies, stakeholders can use more expressive queries, advanced reasoning, and seamless linking to external knowledge bases. Prior research in this space has demonstrated the potential for IFC-to-LBD conversion to address interoperability gaps [4, 6, 16, 17], but few solutions provide a comprehensive methodology spanning standard ontologies that also includes custom damage and domain concepts.

The LifeMACS (life-cycle methodology for the assessment of existing concrete structures) project aims, among other things, to address these challenges by developing a comprehensive framework to aid decision-making in facility management and structural health monitoring [12, 18]. By integrating data from inspections and sensor measurements, the project enables a deep understanding of a structure's performance using Bayesian material degradation and damage evolution models [18], as well as Finite Element (FE) models for structural performance and temporal degradation prediction [19]. These models lead to performance-based optimization of preventive as well as predictive maintenance strategies [20]. The project involves a multidisciplinary team of researchers and industry partners, ensuring the development of a robust, complete, and practical solution.

In this paper, a Linked Data framework is presented that complements existing IFC workflows by incorporating explicit semantic modeling of structural damage, sensor measurements, and other life-cycle parameters—aligned with the needs of LifeMACS. Our approach extends and integrates standard ontologies (BOT [7], SSN/SOSA [21], OPM [8], OMG/FOG [22, 23] or GeoSPARQL [24], OWL-Time, DOT [10]) as well as introduce new concepts in the custom LifeMACS Ontology (LFM). By illustrating the method through a reinforced concrete bridge case study, we show how IFC-to-LBD translation simplifies complex data exchange while enabling deeper statistical analyses for predictive maintenance and asset management. IFC-to-LBD approaches already exist [5, 25], but are out of the box not aligned to the custom ontologies in the project, thus a custom solution was created. Ultimately, this paper contributes a novel, ontology-driven methodology for harnessing Bayesian modeling based on digital building models, reinforcing the broader role of Linked Data and other SWT in the LCM phase of construction.

2. Methodology

2.1. Overall Architecture

The proposed data integration framework adopts a layered architecture designed to transform IFC files into an RDF representation, while accommodating domain-specific extensions required for structural health monitoring and facility management. Figure 1 illustrates the conversion workflow which includes: (1) the IFC data ingestion and processing module, (2) the semantic mapping and ontology alignment steps, and (3) the linked data and query interface. This structure ensures that geometric and semantic information from IFC is retained, processed into specialized ontological constructs, and made available for analysis or integration with external data sources.

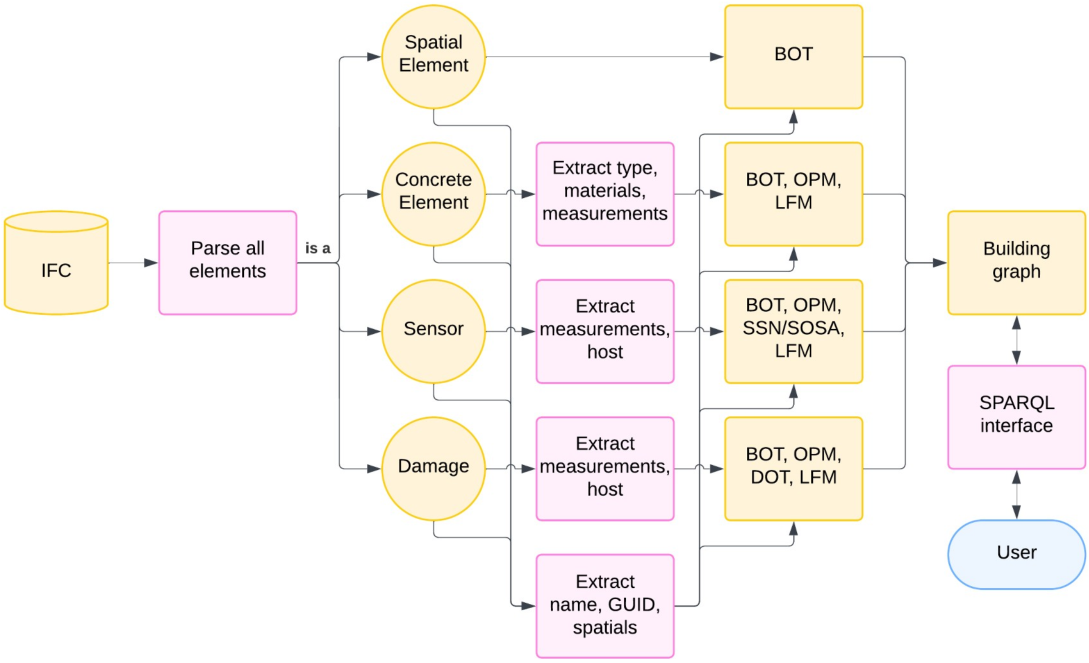

Figure 1: The overall workflow

In the first step, IFC files are collected from existing BIM-based workflows. Then the parser identifies relevant IFC entities (e.g., IfcWall, IfcBeam, IfcSlab, IfcSensor...), extracts the required attributes, and prepares them for conversion. If necessary, partial geometry processing could be performed to determine high-level spatial information, such as element boundaries or positioning, by using for example the GeoSPARQL [24] or OMG/FOG ontologies [22, 23, 26].

The second step addresses the semantic alignment challenge by mapping extracted IFC data to suitable vocabularies and ontologies. This process employs domain ontologies including BOT [7, 27], SOSA/SSN [21], and DOT [10]. Where no suitable classes or properties are available in existing ontologies, custom extensions are defined using the LifeMACS Ontology (LFM) to represent aspects such as Bayesian uncertainties and concrete properties. The mapping engine implements rule-based transformations that convert IFC identifiers and relationships into corresponding triples, ensuring consistent naming conventions and stable Uniform Resource Identifiers (URIs) across the dataset as well as keeping the one-to-one connection to the original IFC object via its global ID (GUID).

The third step stores and serves the resulting RDF graphs through a triple store or linked data platform. This component provides interfaces for data querying and manipulation (e.g., via SPARQL), supporting integrations with other semantic resources. For example, if structural monitoring data—such as strain and temperature measurements—reside in separate repositories, alignment with SOSA properties enables cross-dataset inference and retrieval.

By combining separate data ingestion, ontology alignment, and linked data interaction, this architecture accommodates evolving domain requirements and facilitates future extensions. In particular, time-varying data or probabilistic model outputs were incorporated without extensive modifications to the existing IFC data or underlying ontological models. This approach thus creates a scalable and interoperable foundation for integrating BIM-based geometry and semantic data with specialized workflows in asset management and structural health monitoring.

2.2. Ontology Stack and Custom Extensions

The LifeMACS (LFM) ontology adopts a modular design by importing and extending a series of widely used ontologies to capture both general and specialized aspects of structural health monitoring, building topology, and facility management. As shown in the ontology definition, the following key ontological components are included:

Building Topology Ontology (BOT) provides a concise hierarchy for built-environment concepts, modeling relationships such as adjacency and containment. In the LifeMACS ontology, elements representing physical infrastructure are aligned with bot:Element.

Ontology for Properties and Materials (OPM) is employed to capture material characteristics and property states. In the LifeMACS ontology, the object property lfm:hasPropertyState references opm:PropertyState, enabling time-dependent descriptions of evolving material or structural properties.

Semantic Sensor Network and Sensor, Observation, Sample, and Actuator (SSN/SOSA) are used to describe sensor networks, their observations, and more generally measurements or inspections done. The LifeMACS ontology extends this framework through properties including lfm:hasUncertainty.

Damage Topology Ontology (DOT) is imported to represent damage phenomena, such as cracking or spalling. Instances of dot:Damage can be linked to specific structural components using lfm:hasDamageAssessment.

OWL-Time supplies constructs for representing temporal concepts, such as intervals using time:Interval and specific time points via time:Instant. The LifeMACS ontology references for example time:Interval through properties like lfm:hasValidityPeriod or lfm:hasTimeInterval.

Domain-Specific Extensions (LifeMACS Ontology LFM). Classes such as lfm:BayesianModel and lfm:DegradationModel capture the computational models used for predicting structural performance, while lfm:Parameter and lfm:ProbabilityDistribution enable descriptions of uncertainty in material or model parameters. Furthermore, properties such as lfm:hasIfcRepresentation support linking each LFM individual to its corresponding IFC entity. Additional relationships like lfm:hasFiniteElementModel and lfm:hasCorrosionLevel, ensure that specialized engineering workflows and advanced simulation models are integrated into the overall knowledge graph. The full ontology is available on GitHub[1], as well as an overview of the classes and properties[2].

2.3. IFC-to-TTL Conversion Pipeline

The conversion pipeline from IFC to RDF leverages a Python-based workflow, based on Figure 1, that combines the IfcOpenShell library[3] for parsing IFC data with the RDFLib framework for creating and managing RDF triples. The pipeline is composed of the following sequential stages:

2.3.1. Initialization of Namespaces and Graph Context

The pipeline initializes an empty RDF graph and binds the prefixes mentioned earlier in the ontology stack.

2.3.2. Spatial Structure Extraction

The first task is to extract the hierarchy of spatial entities (e.g., sites, buildings, storeys, and spaces) from the IFC file. In the provided implementation, IfcSite, IfcBuilding, IfcBuildingStorey, and IfcSpace

[1] https://github.com/CedricDriesen92/LifeMACS_LD/blob/main/LFM.ttl
[2] https://github.com/CedricDriesen92/LifeMACS_LD/blob/main/LFM.txt
[3] https://ifcopenshell.org/

elements are iterated to create corresponding instances of BOT concepts (e.g., bot:Site, bot:Building, bot:Storey, and bot:Zone). Relationships such as bot:hasBuilding, bot:hasStorey, and bot:hasZone are then assigned based on the IFC decomposition structure. This process forms a backbone that captures the building's topology and enables downstream steps to attach new entities (e.g., building elements or sensors) to the appropriate location.

2.3.3. Building-Element Conversion

Following the creation of the spatial hierarchy, the pipeline identifies the relevant concrete structural components in the model, such as beams, columns, slabs, and walls. Each element is classified under bot:Element and further specialized to lfm:ConcreteElement to indicate domain relevance for reinforced concrete structures. Where available, the pipeline retrieves references to associated materials (e.g., IfcMaterial) and stores them as property states within the OPM framework (opm:PropertyState). These associations allow explicit representation of each element's material composition and extend potential analyses with domain-specific attributes (e.g., corrosion resistance). Values that have Bayesian properties, such as an uncertainty distribution, are tackled by the LFM ontology via lfm:hasProbabilityDistribution, lfm:mean, lfm:standardDeviation… Finally, each converted entity (including sensor and damage entities) references the original IFC GUID via lfm:hasIfcGlobalId, enabling round-trip interoperability with authoring tools.

2.3.4. Sensor Integration

Sensor data stored as IfcSensor objects are converted to corresponding SOSA classes, labeling each sensor instance with sosa:Sensor. The pipeline also checks IFC property sets for additional metadata (e.g., SensorID, current readings), which is used to create sosa:Observation triples. Temporally relevant data are further annotated using constructs from the Time Ontology (time:Instant), enabling queries about when a measurement was taken. Finally, Bayesian concepts are handled similar to the building elements. This linkage of sensor readings with Bayesian properties to specific building elements or zones supports real-time structural health monitoring use cases.

2.3.5. Damage Modeling

The final conversion step addresses damage information that may be embedded in IfcBuildingElementProxy objects. Properties such as crack length, crack width, and crack depth are detected from relevant IFC property sets. A new dot:Damage instance is generated for each identified defect, and the pipeline associates that instance with its host element through a property such as lfm:hasDamageAssessment. Uncertainties and other Bayesian properties are handled similar to the building elements and sensors. By adopting DOT concepts and linking them to BOT elements, the pipeline preserves both the location and nature of each damage instance.

2.3.6. Output Generation

Once the hierarchical, elemental, sensor, and damage information has been converted, the pipeline serializes the complete RDF graph to Turtle format. This final output may be stored locally for offline analysis or ingested into a triple store for SPARQL-based queries. The serialization step ensures that geometry, topology, and domain-specific attributes are accessible within a single, standards-compliant graph.

2.3.7. Conclusion

Through these discrete stages, the pipeline maintains a separation of concerns between data parsing, semantic enrichment, and RDF output. Specialized Python classes manage the consistency of URIs, while modular functions handle conversions for spatial structure, building elements, sensors, and damages in a reproducible manner. In this manner, the resulting RDF representation reflects both standard building information (in line with IFC and BOT) and specialized life-cycle aspects (e.g., crack measurements, time-stamped sensor observations, probability distributions), allowing for flexible queries and analytics in structural health monitoring and facility management contexts. The full code is available on GitHub[4].

3. Experimental Setup and Evaluation

This section outlines the experimental procedure adopted for converting IFC data to a Linked Data representation within a real-world context. The evaluation focuses on a reinforced concrete bridge case study, selected for its representative structural configuration, active monitoring regime, and the presence of measurable deterioration processes. After describing the test environment and data preparation steps, key outcomes of the IFC-to-LBD conversion are presented.

3.1. Case Study: Reinforced Concrete Bridge

The experimental validation centered on a Belgian mid-20th-century reinforced concrete bridge known to exhibit multiple forms of damage (cracks, spalling, rebar corrosion…). This structure was constructed in the late 1950s and has been subjected to incremental retrofitting and repairs, making it an ideal candidate for testing the capacity of the Linked Data approach to capture both historical and newly acquired data. Examples of the damage can be seen in Figure 2.

Figure 2: Deterioration of the bridge.

A digital model of the bridge was compiled from historical plans and laser scanning. The resulting IFC file served as the starting point for the IFC-to-LBD conversion pipeline. In parallel, a set of sensor measurements—collected via strain gauges and temperature probes distributed over critical structural regions—provided time-series observations indicative of the bridge's real-time performance.

[4] https://github.com/CedricDriesen92/LifeMACS_LD/blob/main/IFCtoLD.py

Additional inspection data, including detail on crack dimensions and material properties, were collected during visual assessments and stored in specialized property sets within the IFC model. The resulting IFC model can be seen in Figure 3, with the cubes representing the sensors and their coloration showing the current sensor strains.

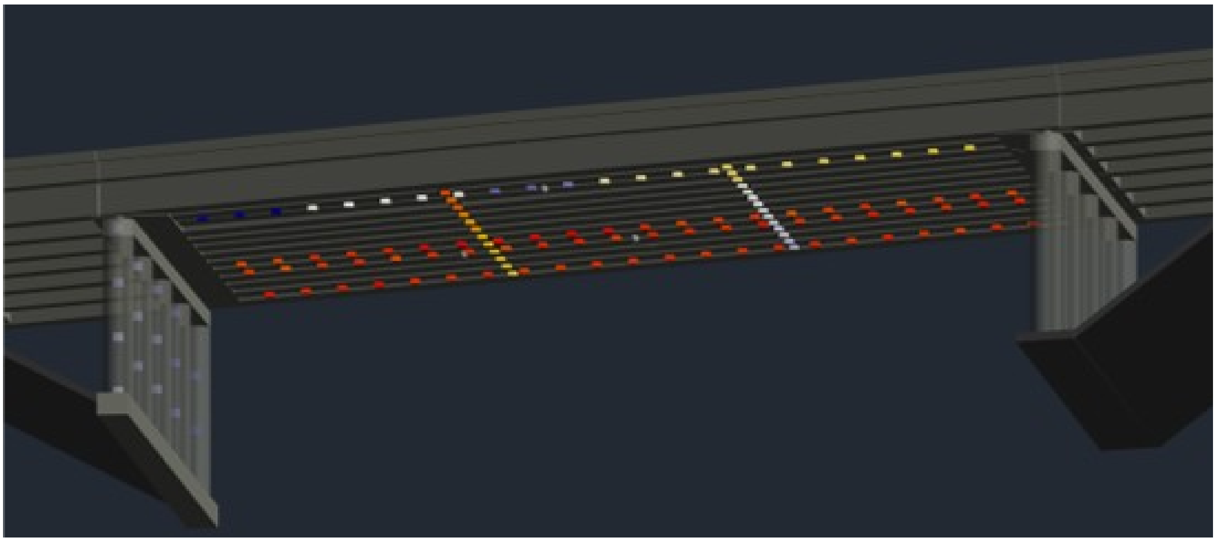

Figure 3: The case study IFC.

3.2. Data Preparation and Pipeline Configuration

Prior to executing the conversion, the IFC model was refined to ensure internal consistency and uniform naming conventions. Elements corresponding to beams, columns, walls, and slabs were verified to include valid standardized property sets specifying both geometry and condition. Sensors in the IFC model were recorded as IfcSensor elements, while defects were cataloged in IfcBuildingElementProxy objects with custom property sets to store crack lengths or spalling area including probability distributions and related concepts, as well as damage types. For simplicity, in this work time dependent data is stored directly in the graph, while for real use cases to avoid making the graph too large a system integration based approach using for example IoT and database workflows should be considered.

The Python-based conversion pipeline (Section 2.3) was configured to:

1. **Extract Spatial Hierarchy** – Identify the site, building (if applicable), and storey relationships and map them to BOT classes (bot:Site, bot:Building, bot:Storey).
2. **Enrich Concrete Elements** – Classify beams, columns, slabs… as lfm:ConcreteElement for subsequent domain-specific queries (e.g., corrosion-level checks, Bayesian model parameters).
3. **Capture Sensor Observations** – Convert IfcSensor instances to sosa:Sensor and generate related sosa:Observation individuals, each annotated with the relevant timestamp and Bayesian properties.
4. **Link Damage Instances** – Detect damage property sets (e.g., crack length, crack depth…) with and convert them to DOT-based damage classes (dot:Damage) with inclusion of Bayesian parameters. The property lfm:hasDamageAssessment was used to associate structural components with specific defect instances.
5. **Maintain Link to IFC** – A property lfm:hasIfcGlobalId is added to all concrete, sensor, and damage elements, with the property value linked to the IFC GUID.

3.3. Conversion Results and Observations

Following the pipeline execution a graph is returned reflecting both the high-level spatial structure (site, storeys, spaces) and detailed elements (concrete sections, sensor data, damage descriptors, uncertainty distributions). Initial inspection of the generated graph indicated that the semantic classes and properties accurately mirrored the specified ontology design.

Spatial concepts were properly instantiated as bot:Site, bot:Building, bot:Storey, and bot:Zone, with containment relationships (e.g., bot:hasStorey) ensuring navigability across levels.
Beams, columns, and slabs retained standard bot:Element membership while also inheriting domain-specific traits via lfm:ConcreteElement.

Each recorded damage instance was expressed as a dot:Damage, carrying quantitative attributes (length, width...) organized under opm:PropertyState as either standard values or uncertainty distributions as well as receiving the required DOT properties such as damage type. This arrangement allowed queries linking defective areas to material states and inspection time frames.

A total of 274 IfcSensor elements produced sosa:Sensor objects, enabling creation of corresponding sosa:Observation instances with distinct time stamps. Observations included (temperature corrected) strain or temperature values tied to specific concrete elements or zones, permitting time-based analyses of structural response, as either values or uncertainty distributions.

4. Results and Discussion

This section interprets the outcomes described in Section 3, analyzing how the proposed pipeline and ontology stack enhance data usage beyond conventional IFC-centric workflows. The discussion situates these observations within the broader context of Linked Data adoption in the AEC industry, both the advantages as well as the limitations.

4.1. Example RDF output and SPARQL query

Following the pipeline execution, a graph of over 4600 RDF triples was generated from the IFC file of the case study bridge, available for viewing on GitHub[5] and exported from Revit 2024 in the IFC 4.3 format. Below in Listing 1 is a short excerpt from this graph illustrating how a damage element might be modeled in Turtle. Suppose an IfcBeam with GUID “0VwJW5Qw3E2uMg8ZhXPz5A” has a damage with GUID “1FsUCh_1n6AAe2OZ$QBHTi”, with crack length stored as a normal distribution with a mean of 17.2 mm and standard deviation of 1.2 mm.

[5] https://github.com/CedricDriesen92/LifeMACS_LD/blob/main/W20.ttl

```
# Concrete element hosting a damage
lfm:element_b9a2457e-24dd-4abc-b91d-50767b1560bc
  a bot:Element, lfm:ConcreteElement ;
  lfm:hasIfcRepresentation “IfcBeam” ;
  lfm:hasIfcGlobalId "0VwJW5Qw3E2uMg8ZhXPz5A" ;
  # Indicate that this element has a damage instance (crack)
  lfm:hasDamageAssessment lfm:damage_3f1dac7a-6a5b-471b-b72e-a847716a2301 .

# Damage instance
lfm:damage_3f1dac7a-6a5b-471b-b72e-a847716a2301
  a dot:Damage ;
  rdfs:label " crack_1" ;
  lfm:hasIfcGlobalId "1FsUCh_1n6AAe2OZ$QBHTi" ;
  # Instead of a single numeric crack length, we store a property state referencing a probability distribution
  lfm:hasPropertyState lfm:property_state_9e5d8ba6-87cc-49f2-a719-5e35c315f711 .

# Property state holding the distribution info
lfm:property_state_9e5d8ba6-87cc-49f2-a719-5e35c315f711
  a opm:PropertyState ;
  opm:propertyName "Crack Length" ;
  prov:generatedAtTime “2024-07-23T12:00:00Z”^^xsd:dateTime
  # We link to a NormalDistribution instance
  lfm:hasProbabilityDistribution lfm:normalDist_32c68ab0-2f14-4c94-9d3f-0aef47b10f06 .

# The NormalDistribution instance
lfm:normalDist_32c68ab0-2f14-4c94-9d3f-0aef47b10f06
  a lfm:NormalDistribution ;
  lfm:mean "17.2"^^xsd:float ;
  lfm:standardDeviation "1.2"^^xsd:float .
```
Listing 1: An example of a damage element in Turtle following the LifeMACS ontology stack

Querying this graph is done by SPARQL query. For example, the query in Listing 2 retrieves all damages longer than 15 mm at $p < 0.05$ for normal distributions, as well as their hosts and other useful properties:

```
SELECT ?element ?elementGuid ?damage ?damageGuid ?meanValue ?stdValue ?lower95
WHERE {
  ?element a lfm:ConcreteElement ;
           lfm:hasIfcGlobalId ?elementGuid ;
           lfm:hasDamageAssessment ?damage .

  ?damage a dot:Damage ;
          lfm:hasIfcGlobalId ?damageGuid ;
          lfm:hasPropertyState ?ps .

  ?ps a opm:PropertyState ;
      opm:propertyName "Crack Length" ;
      lfm:hasProbabilityDistribution ?dist .

  ?dist a lfm:NormalDistribution ;
        lfm:mean ?meanValue ;
        lfm:standardDeviation ?stdValue .

  # Calculate 95% lower bound for damage length = mean - 1.645 * std
  BIND(xsd:float(?meanValue) - 1.645 * xsd:float(?stdValue) AS ?lower95)

  # Filter to return only those cracks whose lower 95% bound is > 15 mm
  FILTER(?lower95 > 15)
}
```
Listing 2: A SPARQL query to retrieve damage elements passing a certain uncertainty threshold

Another example is the SPARQL query in Listing 3, designed to retrieve all latest sensor measurements along with their structural element hosts. Results of a similar query including e.g. spatial coordinates, material properties, uncertainty values… could be fed to a structural FEM-tool.

```
SELECT ?structuralElement ?elementType ?sensorId ?measurementType ?value ?unit ?timestamp
WHERE {
  # Get structural elements with sensors
  ?structuralElement a lfm:ConcreteElement ;
                 rdf:type ?elementType ;
                 lfm:hasSensor ?sensor .

  # Get sensor details
  ?sensor sosa:hasId ?sensorId ;
       sosa:observes ?measurementType ;
       sosa:madeObservation ?observation .

  # Get the observation data
  ?observation sosa:hasResult ?result ;
           sosa:resultTime ?timestamp .
  ?result sosa:hasSimpleValue ?value ;
       sosa:hasUnit ?unit .

  # Subquery to get only the latest measurement for each sensor
  {
    SELECT ?sensor (MAX(?time) as ?timestamp)
    WHERE {
      ?sensor sosa:madeObservation ?obs .
      ?obs sosa:resultTime ?time .
    }
    GROUP BY ?sensor
  }
}
ORDER BY ?structuralElement ?measurementType
```
Listing 3: A SPARQL query to retrieve the latest sensor measurements

Beyond these simple examples one can imagine that, for example, automatic queries of the sensor strain values could lead to automated warnings sent out if certain simulated uncertainty thresholds are crossed.

4.2. Advantages over Traditional Workflows

The experimental findings, such as the examples preceding this section, demonstrate that migrating from a standard IFC file to a Linked Data representation yields several tangible benefits. First, elements such as IfcBeam or IfcColumn become more than static geometric objects once mapped to lfm:ConcreteElement and annotated with domain-specific properties (e.g., damage state, corrosion level, sensor readings). This enriched semantic model permits more direct and flexible queries—such as retrieving all structural members with high corrosion levels, having sensor readings above a defined strain threshold, or crack lengths exceeding a specified limit at a certain level of confidence—that would otherwise require substantial custom coding or a more complicated workflow.

4.2.1. Improved Interoperability and Integration

By aligning IFC concepts with recognized ontologies the resulting graph supports cross-platform data sharing and integration with other Linked Data resources (e.g. the LifeMACS buildingSMART Data Dictionary[6] or other external knowledge bases could easily be implemented).

[6] https://search.bsdd.buildingsmart.org/uri/bw/LM

In the LifeMACS project, among multiple simulation tools (e.g., Bayesian calculation software, finite element packages…), ease-of-use is also increased. Instead of relying on ad hoc code or intermediate export/import routines, these external tools only need to reference well-defined SPARQL endpoints for the single source of truth graph, retrieving exactly the data required.

4.2.2. Addressing Time-Dependent and Probabilistic Aspects

Time-based analysis and probabilistic modeling pose particular challenges when using traditional IFC data structures, which are typically static snapshots of an asset. The proposed ontology and pipeline mitigate these challenges by employing properties from OWL-Time to record observation timestamps, and by introducing custom classes (e.g., lfm:ProbabilityDistribution, lfm:BayesianModel, lfm:NormalDistribution…) to handle stochastic parameters. This opens a pathway for explicitly linking real-time sensor updates or evolving crack measurements to uncertainty models, supporting more robust lifecycle assessments.

4.2.3. Opportunities

While the present case study focused primarily on strain and temperature sensors, the semantic layer is readily extensible to incorporate other data feeds, such as acoustic emission sensors or advanced image-based inspection. Each can be natively represented in SOSA/SSN to ensure standardized observations, procedures, and results.

When integrated into a triple store, the previously mentioned capabilities enable structural engineers and facility managers to carry out more complex investigations (e.g., analyzing damage growth over time in tandem with stress variations) without leaving a single data environment. This is a key advantage over file-based IFC workflows, which often compel stakeholders to use domain-specific applications that can only exchange partial datasets.

4.3. Practical Performance and Usability Considerations

The prototype implementation successfully handled the 10.0 MB bridge IFC data[7] within a runtime of less than 0.1 s on a system with an Intel® i7-13700, with the result being returned in TTL format[8]. However, performance could degrade for significantly larger and more complex models, prompting the need for potential optimizations (parallel parsing, incremental conversion).

4.4. Limitations and Future Opportunities

Despite the clear advantages this methodology presents, a number of limitations remain:

1. **Geometry Representation**
   Although basic spatial constructs (building, storey, zone) have been aligned with BOT, detailed 3D geometry is still stored in the source IFC file. More sophisticated handling, either via other ontologies such as GeoSPARQL [24] or OMG/FOG [22, 23] or more thorough IFC connection, may be necessary for domain applications requiring advanced spatial queries.
2. **Ontology Completeness**
   The LifeMACS ontology currently emphasizes structural and damage-related concepts specific to concrete structures. Adapting or expanding this framework to other domains (e.g., mechanical/electrical systems, occupant comfort) would require further ontology refinement and potentially additional alignments with industry standards.
3. **Data Quality and Governance**

[7] https://github.com/CedricDriesen92/LifeMACS_LD/blob/main/W20.ifc
[8] https://github.com/CedricDriesen92/LifeMACS_LD/blob/main/W20.ttl

As with any semantic integration, the pipeline's efficacy depends on the accuracy and consistency of input data. Missing or mislabeled attributes in IFC can lead to incomplete RDF graphs. Implementing systematic validation checks or "pre-flight" data cleanup routines is recommended to ensure semantic accuracy.

4. **Scalability for Large Infrastructure Networks**
   While the present approach showed feasibility at the level of a single bridge, major infrastructure owners may oversee hundreds of structures. Future work should investigate the performance and user workflow implications of simultaneously processing queries on multiple transformed IFC models.

5. Conclusions and Outlook

The results reaffirm that Linked Data techniques, when properly integrated with IFC workflows, can substantially enhance the management of complex infrastructure data. Bridging geometry and semantically rich domain ontologies allows asset managers to track and query real-time condition indicators, plan maintenance interventions more effectively, and align with state-of-the-art Bayesian modeling and simulation approaches. Although practical hurdles remain (e.g., geometry complexities, adoption by industry professionals), the positive feedback from experts in the LifeMACS project reinforces the potential of IFC-to-LBD integration for real-world SHM and facility management scenarios.

Building on these findings, future research and development could focus on improving geometry serialization into RDF, refining domain-specific ontologies for complementary use cases (e.g., steel or masonry structures), and scaling the solution for large-scale infrastructure portfolios. Continued collaboration with industry stakeholders will be crucial to ensuring that these semantic advancements can be seamlessly integrated into day-to-day engineering and management workflows, ultimately creating a more efficient, interoperable, and intelligent future for AEC data management.

Acknowledgements

The research was done within the FWO SBO project LifeMACS: Multi-layer Bayesian life-cycle Methodology for the Assessment of existing Concrete Structures, supported by FWO-Flanders (FWO-SBO project S001021N).

Declaration on Generative AI

During the preparation of this work, the author used Google Gemini in order to help with grammar/spelling checks, and rewording. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

1. Eastman, C.M., Teicholz, P.M., Sacks, R., Lee, G.: BIM handbook: a guide to building information modeling for owners, managers, designers, engineers and contractors. Wiley, Hoboken, New Jersey (2018)
2. Volk, R., Stengel, J., Schultmann, F.: Building Information Modeling (BIM) for existing buildings — Literature review and future needs. Autom. Constr. 38, 109–127 (2014). https://doi.org/10.1016/j.autcon.2013.10.023
3. Koeleman, J., Ribeirinho, M.J., Rockhill, D., Sjodin, E., Strube, G.: Decoding digital transformation in construction. McKinsey (2019)
4. Hernández, J.L., Lerones, P.M., Álvarez, S., Bonsma, P., van Delft, Deighton, R., Braun, J.-D.: An IFC-based interoperable framework for building linked-data. Presented at the LDAC (2018)
5. Pauwels, P., Terkaj, W.: EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Autom. Constr. 63, 100–133 (2016). https://doi.org/10.1016/j.autcon.2015.12.003
6. Pauwels, P., Zhang, S., Lee, Y.-C.: Semantic web technologies in AEC industry: A literature overview. Autom. Constr. 73, 145–165 (2017). https://doi.org/10.1016/j.autcon.2016.10.003
7. Rasmussen, M.H., Lefrançois, M., Schneider, G.F., Pauwels, P.: BOT: The building topology ontology of the W3C linked building data group. Semantic Web. 12, 143–161 (2020). https://doi.org/10.3233/SW-200385
8. Rasmussen, M.H., Lefrancois, M., Bonduel, M., Hviid, C.A., Karlshø, J.: OPM: An ontology for describing properties that evolve over time. Presented at the LDAC (2018)
9. Chamari, L., Petrova, E., Pauwels, P.: A web-based approach to BMS, BIM and IoT integration: a case study. CLIMA 2022 Conf. (2022). https://doi.org/10.34641/clima.2022.228
10. Hamdan, A.-H., Bonduel, M., Scherer, R.J.: An ontological model for the representation of damage to constructions.
11. Peñaloza, R.: Introduction to Probabilistic Ontologies. In: Manna, M. and Pieris, A. (eds.) Reasoning Web. Declarative Artificial Intelligence. pp. 1–35. Springer International Publishing, Cham (2020)
12. Botte, W., Vereecken, E., Caspeele, R.: Random field modelling of spatial variability in concrete – a review. Struct. Infrastruct. Eng. 1–14 (2023). https://doi.org/10.1080/15732479.2023.2248102
13. Artus, M., Koch, C.: State of the art in damage information modeling for RC bridges – A literature review. Adv. Eng. Inform. 46, 101171 (2020). https://doi.org/10.1016/j.aei.2020.101171
14. Xu, S., Wang, J., Wang, X., Wu, P., Shou, W., Liu, C.: A Parameter-Driven Method for Modeling Bridge Defects through IFC. J. Comput. Civ. Eng. 36, 04022015 (2022). https://doi.org/10.1061/(ASCE)CP.1943-5487.0001026
15. Riaz, Z., Parn, E.A., Edwards, D.J., Arslan, M., Shen, C., Pena-Mora, F.: BIM and sensor-based data management system for construction safety monitoring. J. Eng. Des. Technol. 15, 738–753 (2017). https://doi.org/10.1108/JEDT-03-2017-0017
16. Beetz, J., Borrmann, A.: Benefits and Limitations of Linked Data Approaches for Road Modeling and Data Exchange. In: Smith, I.F.C. and Domer, B. (eds.) Advanced Computing Strategies for Engineering. pp. 245–261. Springer International Publishing, Cham (2018)
17. Pauwels, P., McGlinn, K. eds: Buildings and Semantics: Data Models and Web Technologies for the Built Environment. CRC Press, London (2022)
18. Vandecruys, E., Van De Velde, M., Reynders, E., Lombaert, G., Verstrynge, E.: Experimental study on acoustic emission sensing and vibration monitoring of corroding reinforced concrete beams. Eng. Struct. 293, 116553 (2023). https://doi.org/10.1016/j.engstruct.2023.116553
19. Helderweirt, S., Van Den Hende, K., Botte, W., Verstrynge, E., Caspeele, R.: Quantifying Uncertainties in the Performance Prediction of Existing Concrete Structures using an Extended Direct Stiffness Method Approach. Presented at the ICASP14 (2023)
20. Caspeele, R., Van Den Hende, K.: Validation of the harmonized partial factor method for design and assessment of concrete structures as proposed for fib model code 2020. In: Structural Concrete. pp. 4368–4376 (2023)

21. Janowicz, K., Haller, A., Cox, S.J.D., Le Phuoc, D., Lefrançois, M.: SOSA: A lightweight ontology for sensors, observations, samples, and actuators. J. Web Semant. 56, 1–10 (2019). https://doi.org/10.1016/j.websem.2018.06.003
22. Bonduel, M., Wagner, A., Pauwels, P., Vergauwen, M., Klein, R.: Including widespread geometry formats in semantic graphs using RDF literals. Presented at the 2019 European Conference on Computing in Construction July 10 (2019)
23. Wagner, A., Bonduel, M., Pauwels, P., Uwe, R.: Relating geometry descriptions to its derivatives on the web. Presented at the 2019 European Conference on Computing in Construction July 10 (2019)
24. Battle, R., Kolas, D.: GeoSPARQL: Enabling a Geospatial Semantic Web.
25. Bonduel, M., Oraskari, J., Pauwels, P., Vergauwen, M., Klein, R.: The IFC to Linked Building Data Converter - Current Status.
26. Wagner, A., Bonduel, M., Werbrouck, J., McGlinn, K.: Geometry and geospatial data on the web. In: Buildings and Semantics. CRC Press (2022)
27. Schneider, G.F.: Towards Aligning Domain Ontologies with the Building Topology Ontology. (2017). https://doi.org/10.13140/RG.2.2.21802.52169